

# A Cubesat Swarm Approach for Exploration of the Asteroid Belt

Patrick H. Stakem,  
Johns Hopkins University, Whiting School of Engineering.

José Carlos Martínez  
Telecom Engineering PhD. Student,  
University of Granada, Spain.

Vishnu Chandrasenan, PhD  
Telecommunications Program,  
Department of Electrical and Computer Engineering,  
A. James Clark School of Engineering,  
University of Maryland-College Park.

Yash Mittra  
Dept. Of Information Technology,  
Guru Gobind Singh Indraprastha University  
Dwarka, Delhi, India.

(c) 7/17/18

Presented to NASA Goddard Planetary CubeSats Symposium  
August 16-17, 2018.  
NASA, GSFC, Greenbelt, MD.

This paper describes an approach to exploration of the asteroid belt, and discusses the advantages and challenges of using a Cubesat Swarm. This approach was first explored as a multi-student study of an alternative to the Juno Mission to Jupiter. We worked within the mission's size, weight, and power envelopes. Staying within the parameters of the mission, we were able to accommodate 1,000-1U Cubesats. We chose 333 3-U, and explored issues in local control, networking communications, and implementation in Open Source. This was the baseline for defining the Asteroid Swarm Mission. The Mission Parameters are quite different.

Exploring the asteroids requires a diverse and agile system. A swarm of small spacecraft with different capabilities will be used, combining into Teams to address situations and issues discovered in situ. A intelligent swarm solution to resource exploration of the asteroid belt was proposed by Curtis, et al, in 2003. The concept of Cubesats was not advanced enough at the time for the authors to specifically mention them, and explored issues in local control, networking communications, and implementation in Open Source.

This is a global, cooperative, open source project. We did not calculate all of the component Technology Readiness Levels, but do include the heritage of the concepts. This is an ongoing effort.

## **Asteroid Belt**

The Asteroid belt is located between 2.0 AU and 3.2 AU, with Mars being at 1.67 AU. It forms a continuous, sparsely populated ring around the Sun. The average distance between the objects in the belt is large, and solid objects are sparse.

A driver in asteroid belt exploration is the sheer number of things that need to be explored. We can no longer afford to send dedicated single spacecraft missions to all the potential targets in our solar system. Although there are fewer than 10 planets, and less than 200 moons, there are millions of asteroids, mostly in the inner solar system. Each asteroid may be unique, and some may provide needed raw materials for Earth's use. The asteroid belt contains Ceres, the Dwarf planet, and some 750,000 rocks larger than one kilometre in diameter. There are three main classifications: carbon-rich, stony, and metallic.

The radiation environment of the asteroid belt is mostly of solar origin, with some Galactic cosmic rays. There is no overall magnetic field to deflect energetic particles. A lot of the material in the belt is dust sized, and will pose a damage threat to spacecraft.

The physical composition of asteroids is varied and poorly understood. The larger Ceres (referred to as a dwarf planet) appears to be composed of a rocky core covered by an icy mantle, and is thought to harbor organic compounds. 4-Vesta may have a nickel-iron core. It was explored by the Dawn spacecraft in 2012. 10-Hygia appears to have a uniformly primitive composition of carbonaceous chondrite. Many of the smaller asteroids are piles of rubble held together loosely by gravity. Some have moons themselves, or are co-orbiting binary asteroids. The bottom line is, asteroids are diverse. There is even speculation that Ceres may harbor life.

The asteroids are not uniformly distributed. In the asteroid belt, the Kirkwood gaps are relatively empty spots. This is caused by orbital resonance of the asteroids with Jupiter. Orbiting irregularly shaped bodies is challenging, due to the irregular gravity field. This makes station keeping and attitude control a bigger problem.

It has been suggested that asteroids might be used as a source of materials that are rare or exhausted on Earth (asteroid mining) or materials for constructing space habitats and refueling stations for farther missions. Materials that are heavy and expensive to launch from Earth may someday be mined from asteroids and used for in situ space manufacturing. Valuable materials such as platinum and rare and scarce elements may be returned to Earth for a profit.

## **Enabling Technologies**

This section will discuss some of the enabling technologies that will be used in the Cubesat Swarm mission definition. Use of existing technologies from previous missions was a goal. An ops concept will be presented later.

## **Overall Architecture**

In this concept, the Cubesats are the primary payload. The Mothership can be thought of as a very large Cubesat. The architecture is kept as close as possible.

Use of a common hardware bus and software architecture for all swarm members, to the greatest extent possible, is a goal. Only the sensor sets will be unique. A Cubesat model for the hardware, and NASA GSFC's Core Flight Software is baselined. A standard linux software operating environment and database will be used.

Each member of the swarm will be aware visually of other swarm members in close proximity. This will be facilitated by having the Mothership as the center of the coordinate system. It will determine its position by celestial navigation. The Cubesats will have a similar capability. The mothership will maintain, as part of its onboard database, the location of all other members. It will also monitor for pending collisions and warn the participants. There will be rules concerning how close swarm members can get to each other, a virtual zone of exclusion. All Earth-based interaction with the swarm will be through the Mothership. Due to varying communication delays, operation of the swarm by teleoperation from Earth is not feasible. The Swarm could be on the opposite side of the Sun from the Earth for extended periods. This is addressed by building autonomy into the system, and a large amount of non-volatile storage will be included for science data.

In this concept, the Cubesats are the primary payload. The Mothership can be thought of as a very large Cubesat. The architecture is kept as close as possible.

Use of a common hardware bus and software architecture for all swarm members, to the greatest extent possible, is a goal. Only the sensor sets will be unique. A Cubesat model for the hardware, and NASA GSFC's Core Flight Software is baselined. A standard linux software operating environment and database will be used.

Each swarm member will be equipped with one or more cameras, not only for target investigation, but also for observing the position and relative motions of other swarm members.

Using standard linux clustering software (Beowulf), the Mothership and undeployed swarm members will be able to form an ad-hoc cluster computer to process science data in-situ. Within the Mothership, LAN-based Mesh network software will be used. The Mothership's main computer will be a

Raspberry-Pi based cluster. For the Jupiter case, we baselined Juno's rad-hard RAD-750, which is limited in computational resources.

Complexity in a system generally derives from two parameters, the number of units, and the number of interactions. A swarm of Cubesats is complex, compared to a single spacecraft. This is somewhat balanced by the relative simplicity of the units, their standardization, flexibility, and redundancy.

## ***Batteries and solar panels***

The advanced batteries and solar panels on the ongoing Juno mission at Jupiter are functioning well. These would then be a candidate for the Swarm Mothership. Due to technology advances, solar cells can now be used out to 5AU. The Mothership will have large arrays, but the individual Cubesats have limited area. They will be deployed fully charged. One area of interest is a deployable fabric solar panel.

At the distance of the asteroid belt, the solar constant ( $\text{kw}/\text{meter}^2$ ) is about  $\frac{1}{4}$  of that at Earth's distance from the sun.

## **Cubesats**

The Cubesats will be a mix of 3U and 6U in size (a U-Unit is 30x10x10 cm), and follow the GSFC-defined PiSat architecture with a Raspberry-Pi flight computer, running NASA/GSFC's CFS/CFE flight software. Different instrumentation will be included on different Cubesats, using common platforms and buses. These will be deployed by the Mothership as required, to observe and collect data on targets of interest. A lander-Cubesat with additional payload and a propulsion system can also be included.

## **Dispenser/Mothership**

The Mothership will be built with standard aerospace products with a mission heritage. We expect to be able to use the same batteries and solar panels from the Juno mission, since the Mothership will be roughly the same size, defined by the size of the launch vehicle shroud. The X-band transceiver on Juno would be a candidate for the Earth link. The carrier is designed to be modular and adaptable. It uses standard PPOD Cubesat deployment mechanisms, oriented radially. Standard p-pod Cubesat dispensers are baselined, but the affects of long term storage of the Cubesats in the dispensers in space must be carefully considered. The effects of cold welding during the transit and storage time needs to be evaluated.

Unlike Earth Cubesat missions, the Cubesats going to the Belt can have their own propulsion. The big limiting factor for them is electrical power. They can't carry large solar arrays. Dispersed from the carrier fully charged, they will operate as long as they can. The electronics and software will be optimized to minimize power usage. More advanced solar arrays, possible fabric-based arrays that will serve double duty as a solar sail, could solve this limitation.

The Mothership provides cloud services to the swarm. It is a store-and-forward node, and the communications relay to Earth. It provides Swarm control, monitoring, and task assignment.

The Mothership will have a bi-propellant engine for orbit and cruise adjustments, and a monopropellant

system for attitude control and reaction wheel momentum unloading.

## **Cubesat Swarm**

This section describes a different approach to exploration with collections of smaller co-operating systems that will combine their efforts and work as ad-hoc teams on problems of interest.

Swarm behavior is based on the collective or parallel behavior of homogeneous systems. It is implemented as multiple co-operating software agents. This mimics collective behavior, modeled on biological systems. Examples in nature include migrating birds, schooling fish, and herding sheep. A collective behavior emerges from interactions between members of the swarm, and the environment. The resources of the swarm are organized dynamically. Swarm behavior is a group attribute across the swarm. It relies on continuous co-ordination of behavior. Each member makes stochastic choices. There is a lack of authority in the Swarm, each member making local choices based on local conditions, based on a global rule set. Example rules are separation, avoid crowding; alignment, do what other swarm members are doing; and cohesion, move to the center of mass of the swarm. Swarm activity can be chaotic or orderly. Emergent, un-planned behaviors are seen in implemented systems.

In Swarm systems, the key issues are communication between units, and cooperative behavior. The capability of individual units does not much matter; it is strength in numbers. Self-organizing behavior emerges from decentralized systems that interact both with members of the group, and the environment. Swarm intelligence is an emerging field, and swarm robotics is an active research topic.

The swarm will be agile, able to respond to targets of opportunity, when they arise. Flexibility, and the ability to respond locally to unplanned events will be part of the architecture. A Swarm is an implementation of a multi-agent system, where the agents are implemented in program code.

A constellation of 50, 2U and 3U Cubesats was deployed in orbit in 2015. Some were released from the ISS, in coordination with a rocket launch. They collected and telemetered data on the lower thermosphere. This was not a swarm per se, but rather 50 units acting on their own, reporting back to their home institutions. Universities around the world participated in the project.

The data came from the region below 85 kilometers, which has enough of an atmosphere to impede spacecraft. The Cubesats collected data as long as they could, as they were reentering the atmosphere. At these altitudes, the rarefied atmosphere can reach temperatures of 2,500 degrees C. It is also a region where the dynamics are controlled by atmospheric tides, themselves controlled by diurnal heating and cooling. The member Cubesats used onboard processing to reduce downlink bandwidth.

## **Avionics Suite**

Both the Mothership and the Cubesats will baseline the GSFC PiSat software and hardware architecture for the flight computers. The Cubesats will use a single unit, and the Mothership will have a 16-unit cluster. Non-deployed Cubesats in the Mothership will be able to participate in the clustering, using the Mothership's internal networking infrastructure. The Mothership will be able to power up and attach selected additional units for particularly computer-intensive tasks.

The Raspberry Pi-3 is a very capable processor. An earlier model was tested to operate to 150 kRad

Total Ionizing Dose, with only the loss of several unused I/O interfaces. The major source of radiation at the destination will not be trapped particles, but rather ionizing cosmic rays of galactic origin. These are energetic, but sparse. The cluster computer will be enclosed in the nose of the mothership.

A Raspberry Pi-3, requires 3.26 watts of power. It is quad core, operating at 1.4 GHz. It is a 64-bit machine, with 1 gigabyte of ram, and can achieve 2451 MIPS. It has a dedicated Graphics Processing Unit-based video pipeline that can handle 2D DSP. That is supported by the Open-GL software.

## ***Compute cluster of convenience***

Using a variation of the Beowulf clustering software and the communications infrastructure of the Mothership, the Cubesats awaiting deployment can be linked into a Compute cluster configuration. Each compute node will have the Beowulf software pre-loaded as part of its Linux operating system.

Beowulf was developed to provide a low cost solution to linking commodity pc's into a supercomputer. The approach has been applied to clusters of small architectures such as those that serve as flight computers for Cubesats. Several 64-node Pi clusters have been demonstrated in the Earth environment.

The Beowulf cluster is ideal for sorting and classifying data; an example application is the Probabilistic Neural Network. This algorithm has been used to search for patterns in remotely sensed data. It is computationally intensive, but scales well across compute clusters. It was developed by the Adaptive Scientific Data Processing (ASDP) group at NASA/GSFC. The program is available in Java source code.

The first Beowulf cluster to be flown in space was built from twenty 206-MHz StrongARM (SA1110) processors, and flew on the X-Sat, Singapore's first satellite. The performance was 4,000 MIPS. The cluster drew 25 watts. The satellite was a 100 kg unit, 80 CM cube. The cluster was used because the satellite collected large amounts of image data (80 GB per day), most of which was not relevant to the mission. An onboard classification algorithm selected which images would be downloaded. For example, cloudy images were discarded, since land images of Singapore were of interest.

In a cluster, there is always a trade-off of computation and communication, with power draw. This will be monitored and adjusted by the cluster itself.

## **Surface Lander**

In the surface exploration scenario, a 6U Cubesat will serve as a local Mothership. It will be able to detach and deploy the lander vehicle. The orbiting Cubesat can provide a “gods-eye” view, to target locations for direct exploration. It will also serve as a communications relay. The main problem will not be mobility, but rather avoiding floating off the surface.

The ground-based unit will implement prox-ops in a more leisurely fashion, in that motion will be two-dimensional and at low speed. The surface rovers will not be retrieved. The local Mothership may be left at the observation site for additional data, or can be redeployed to an additional target. The location of a lander/rover on the surface will be determined by the orbiting or station-keeping Mothership, with imaging. Standard space communications protocols will be used between the lander and the Mothership, via UHF link.

The Cubesat members will collect observation data on their target, and can conduct radio occultation experiments to better categorize the distribution of particles. They can also conduct synchronized simultaneous observation from multiple observation points of features of interest.

The Mothership is also an observer, with an instrument suite. It can search for magnetic fields, and characterize the charged and energetic particle environment.

## **Communications**

Several approaches to communication with spacecraft at large distances from Earth, and examining other planets, have been defined. The Interplanetary Internet implements a Bundle Protocol to address large and variable delays. Normal IP traffic assumes a seamless, end-to-end, available data path, without worrying about the physical mechanism. The Bundle protocol addresses the cases of high probability of errors, and disconnections. This protocol was tested in communication with an Earth orbiting satellite in 2008. It is derived from the mobile data protocols used with cell towers in terrestrial applications. The Disruption Tolerant Network approach is also a good candidate.

As we get farther from Earth, the Cubesat's small antennas, and relatively low power, means we have to get clever with communications. There will be limited bandwidth. This was the case with APL's Horizon's spacecraft at Pluto – It took more than 16 months to transmit all the data back from the encounter.

The deployed Cubesats will use the Mothership as their communication relay, and not necessarily implement Cubesat-Cubesat direct communications. Cubesat to Mothership will be S-band. It does not necessarily need to implement a delay tolerant protocol, since the Cubesats will be “in the vicinity” of the Mothership. Cubesats will exchange two types of data with the Mothership: primary observational data, and secondary metadata which includes position, localization information, and timing information.

All data will be kept and transmitted in CCSDS format.

## **Communication methodology within the Cubesat network.**

ISL (Inter-satellite link) communication will be achieved on Ultra High Frequency (UHF) links. An inter-satellite communication range of 90km to 100km is viable on UHF within the power output range of 4-5 W. The next challenge is regarding selection of the ideal antenna and communication protocol, keeping in mind the existing power and mobility constraints along with the tradeoff between radio power and communication distance. NASA's Nodes (Network & Operation Demonstration Satellite) mission, similar in structure to the Edison Demonstration of Smallsat Networks (EDSN) mission, deployed a satellite swarm of Cubesats from the ISS to test inter-satellite communication capabilities in 2015. A primary UHF radio was used for crosslink communication, and a further UHF beacon radio was used for transmitting real time health information of the satellite. In addition to this, position, navigation, and tracking information complement the primary data load. The Cubesats, and the Mothership will use software defined radio, implemented on the flight computer.

## Ops concept

During the cruise phase to the Asteroid Belt, the Cubesats are unpowered. Every day or week (tbd), the units are powered on, one at a time, and checked for functionality. The onboard database is updated as required. The results are sent back to the control center on Earth. One advantage of the Mothership is, like the Shuttle, payloads can be tested before deployment. Known bad units will be left in place, or discarded into Space.

Near the desired target location at the Asteroid belt, the Mothership uses its main engine to enter solar orbit with the solar panels oriented to the Sun, and the high gain antenna pointed to the Earth.

After another system check of itself and the Cubesats, the Mothership deploys a series of Cubesat scouts on a reconnaissance mission, to seek out areas of interest. The Mothership deploys Cubesats with broad spectral sensing capabilities. Based on their findings, the Mothership may deploy additional Cubesats with specific instrumentation to the area of interest. (For example, an advanced thermal imager to an area of observed thermal activity). The Cubesats are released in the order of necessity. There will only be one Cubesat per dispenser, so blocking is not an issue.

In the surface exploration scenario, the lander necessarily needs a propulsion system for alignment and touchdown. This would be a cold gas system. It would be possible, but more complex, to implement a sample return. This would involve descent and ascent, return to the Mothership, and rendezvous and docking. It is anticipated that the entire Cubesat with its payload would be returned to Earth, which is simpler than a sample hand-off to a return vehicle.

## Rad-Hard Software

This is a concept that implements routines that check and self-check, report, and attempt to re-mediate. It is an outgrowth of the testing and self-testing of a computers' functionality, with focus on detection of radiation induced damage. We know, for example, that one of the tell-tales for radiation damage is increasing current draw. At the same time, we monitor other activities and parameters in the system. This partially addresses the problem of operating with non-radiation hardened hardware in a high radiation environment. The baseline RaspberryPi has been radiation tested to 150kRad, and was operational at that point.

From formal testing results, and key engineering tools, we define likely failure modes, and possible remediations. Besides self-test, we will have cross-checking of systems. Not everything can be tested by the software, without some additional hardware. First, we use engineering analysis that will help us define the possible hardware and software failure cases, and then define possible actions and remediation. None of this is new, but the approach is to collect together best practices in the software testing area, develop a library of RHS routines, and get operational experience. Another advantage of the software approach is that we can change it after launch, as more is learned, and conditions change.

Rad Hard software runs in the background on the flight computer, and checks for the signs of pending failure from any known cause. The biggest indicator for radiation damage is an increase in current draw. The flight cpu monitors and trends current draw across the swarm, and take critical action such as



a reboot if it deems necessary. The Rad Hard software will keep tabs on memory by conducting continuous CRC (cyclic redundancy checks). One approach to mitigating damage to semiconductor memory is “scrubbing,” where we read and write back each memory locations (being careful not to interfere with ongoing operations). This will be done by a background task that is the lowest priority in the system. Watchdog timers are also useful in getting out of a situation such as a Priority Inversion, or just a radiation-induced bit flip. There will be a pre-defined safe mode for the computer as well. Key state data from just before the fault will be stored. Unused portions of memory can be filled with bit patterns that can be monitored for changes. We must be certain that all of the unused interrupt vectors point to a safe area in the code, so this will be reloaded periodically.

Functions within the RHS include current monitoring as a tell-tale of radiation damage, self-diagnosis suite, spurious interrupt test, memory test(s), checksums over code, data corruption testing, memory scrub, I/O functionality test, peripherals test, stack overflow monitoring, and a watchdog timer. A complete failure modes and effects analysis will be conducted over the flight computer and associated sensors and mechanisms, and this will be used to scope the RHS. The systems will keep and report trending data on the flight electronics. In most cases, the only remediation is a reboot. However, since the units will have identical configurations, the data will be useful to be able to predict pending failures, and to possibly avoid and correct them. This will be used on the Mothership's and on the Cubesat's RaspberryPi-based flight computers. This provides a distributed fault detection and mitigation system, with learning.

We can also choose to implement a small, rad-hard recovery computer, which uses FRAM, which is fairly immune to radiation. The recovery computer would receive heart-beat signals from the cluster members onboard the mothership, and take recovery efforts if they are interrupted. A similar scheme could be used onboard the Cubesats, with little impact on size, weight, power, and cost. This would primarily be used to mitigate latchup.

## **Onboard databases**

Each member of the Swarm is self-documenting. It carries a copy of its Electronic Data Sheet (EDS) description, which will be updated. This defines the system architecture and capabilities, and has both fixed (as-built) and variable entries. The main computer in the Mothership has a copy of all of these, and can get updates by query. The Mothership also has parameters on each unit's state, such as electrical power remaining, temperature, position, etc. One value of the database is, if the Mothership needs a unit with a high resolution imager, it knows what unit that is, and whether it has been deployed or not. If it has been deployed, it will query the unit on its position and health status. Implementing the EDS in a true database has advantages, since the position of the data item in the database also carries information. It also allows the use of off-the-shelf database tools. The individual Cubesats have a “light-weight” version of the database, while the Mothership has a more sophisticated one. All the schema's are the same. The advantage of a formal database is the structure it imposes on the data

There are two parts of the tables, representing static and dynamic data. Static Data represents the hardware and software configuration of the swarm unit. These values are not expected to change during the unit's operation. The Dynamic Data table represents the sensors each particular unit has. These values can change, and the last values will be kept. Satellites will exchange two types of data through

its communication channel: primary observational data, along with secondary metadata which includes position and localization information along with timing information as a part of the EDS during the mission. This approach was prototyped in a previous project.

The Mothership is responsible for aggregating all of the Cubesats' housekeeping and science data, and transmitting it back to Earth. This is also facilitated by the structure imposed by the database. An Open Source version of an SQL database will be implemented. The EDS documents will be in XML.

Data compression can be implemented onboard the mothership , as well as preliminary data analysis for replanning.

### **Fly the Control Center**

The Mothership is the navigation reference point for the Cubesats. It obtains its position with respect to Earth from observation, and ground tracking. There will be times when the Earth is not visible from the Mothership's position, so it will use extrapolation and local observation. During these periods of occultation, and also periods of long one-way light times, the Mothership assumes local responsibility for the Health and Safety of the Swarm members. For this, we will implement Control Center functionality within the Mothership. This will take the form of Ball Brother's COSMOS software. This product addresses traditional system test, integration, and flight needs. An additional software module is needed, essentially a virtual Control System Operator. Using defined rules, the Mothership will make decisions concerning the Swarm Members, to the best of its current knowledge. All of this will be documented and downloaded to the Earth-based control center when communications is re-established. An AI capability will be added to Cosmos, in the form of a virtual flight controller agent. Besides the housekeeping functions, we will implement onboard science planning, responsive to on-site conditions, and targets of opportunity.

The Mothership's primary responsibility is continuance of the Mission. To a degree, the Cubesats are considered expendable. During communications black-outs, observations will continue, and the Mothership will dispense explorers according to pre-defined rules, and based on it's best on-scene judgment. It will also continue to collect observation science data, and engineering data related to health and performance across the swarm members.

### **Wrap-Up**

The paper study of a Cubesat mission for Exploration of the Asteroid Belt has brought together a series of existing concepts, hardware, and software that would allow implementation of the Asteroid Belt mission. The core concepts are to use, to the extent possible, Open Source hardware and software, as well as flight-proven high TRL components from other missions.

A distributed target requires a distributed approach to exploration.

## Bibliography

Alvarez, Jennifer L.; Rice, John R. Samson, Jr., Michael A. Koets. "Increasing the Capability of Cubesat-based Software-Defined Radio Applications," avail: [ieeexplore.ieee.org/document/7500847/](http://ieeexplore.ieee.org/document/7500847/)

Budianu, S.; Engelen, R. T.; Rajan, A. Meijerink; Verhoeven, C. J. M.; Bentum, "The Communication Layer for the OLFAR Satellite Swarm." (2011). avail: [doc.utwente.nl/78092/](http://doc.utwente.nl/78092/)

Challa, Obulapathi N., McNair, Janise "Cubesat Torrent: Torrent-like distributed communications for Cubesat satellite clusters," Military Communications Conference, 2012 , (MILCOM 2012) July 19, 2016. avail: <https://www.researchgate.net/.../261237135>

Challa, Obulapathi N., McNair, Janise "Distributed Data Storage on Cubesat Clusters," Advances in Computing 2013, (3) 3 pp.36-49. avail: <http://article.sapub.org/10.5923.j.ac.20130303.02.html>

Clark, P. E.; et al *BEEES for ANTS: Space Mission Application for the Autonomous NanoTechnology Swarm*, avail: <https://arc.aiaa.org/doi/abs/10.2514/6.2004-6303>.

Cudmore, Alan Pi-Sat: A Low Cost Small Satellite and Distributed Mission Test Program, NASA/GSFC Code 582, avail: <https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20150023353.pdf>

Curtis, S. A. et al "Use of Swarm Intelligence in Spacecraft Constellations for the Resource Exploration of the Asteroid Belt," 2003, Third International Workshop on Satellite Constellations and Formation flying, Pisa, Italy.

Hall, John "maddog"; Gropp, William *Beowulf Cluster Computing with Linux*, 2003, ISBN -0262692929.

Jones, Nicola "Tiny 'chipsat' spacecraft set for first flight," Nature, June 2016, Vol 534. avail: [www.nature.com/news/tiny-chipsat-spacecraft-set-for-first-flight-1](http://www.nature.com/news/tiny-chipsat-spacecraft-set-for-first-flight-1)

Kirkpatrick, Brian, et al *Dynamics and Control of Cubesat Orbits for Distributed Space Missions, Aerospace Corporation, 2015*, avail: [www.ipam.ucla.edu/wp.../Aerospace-Corp-project-description-FINAL.pdf](http://www.ipam.ucla.edu/wp.../Aerospace-Corp-project-description-FINAL.pdf).

Hinchey, Michael G. ; Rash, James L.; Truszkowski, Walter E.; Rouff, Christopher A., Sterritt, Roy *Autonomous and Autonomic Swarms*, avail: [https://ntrs.nasa.gov/search.jsp?R=20050210015 2017-12-20T20:19:24+00:00Z](https://ntrs.nasa.gov/search.jsp?R=20050210015%2017-12-20T20:19:24+00:00Z)

Madni, Mohamed Atef; Raad, Raad; Tubbal, Faisal "Inter-Cubesat Communications: Routing Between Cubesat Swarms in a DTN Architecture," presentation, avail: <https://iCubesat.org/papers/2015-2/2015-b-2-1>.

Macdonald, Malcolm "Advances in Solar Sailing,"2014, Springe Praxis, ISBN-978-3642349065.

McLoughlin, Ian; Bretschneider, Timo; Ramesh, Bharath “First Beowulf Cluster in Space,” Linux Journal, September 2005, Issue #137, article 8097.

Muri, P., McNair, J. “A Survey of Communication Sub-systems for Intersatellite Linked Systems and Cubesat missions”, J. Comm., 7 (2012), pp. 290–308.

Popescu, Otilia, “Power Budgets for Cubesat Radios to Support Ground Communications and Inter-Satellite Links” by (Senior Member, IEEE), Department of Engineering Technology, Old Dominion University, Norfolk, <https://ieeexplore.ieee.org/document/7964683/>

Robson, Christopher, “Comparison of Cubesats, Cubesat Swarms and Classical Earth Observation Satellites in LEO, Canadian SmallSat Conference, 2016, avail:  
<https://canadiansmallsatssymposium2016.sched.com/event/4bOr/comparison-of-Cubesats-Cubesat-swarms-and-classical-earth-observation-satellites-in-leo>

Ruggieri, Marina et al “The Flower Constellation Set and its Possible Applications, Final Report,” avail: [www.esa.int/act](http://www.esa.int/act)

Saks, N. A.J. Boonstra, R.T. Rajan, M.J. Bentum, F. Beliën, and K. van’t Klooster, “DARIS, A Fleet of Passive Formation Flying Small Satellites for Low Frequency Radio Astronomy,” The 4S Symposium, Small Satellites Systems and Services, Madeira, Portugal, May-June 2010. avail:  
<https://ris.utwente.nl/ws/portalfiles/portal/5513988>.

Spangelo, et al “JPL's Advanced CubeSat Concepts for Interplanetary Science and Exploration Missions, Cubesat Workshop,” 2015, California Institute of Technology, JPL.  
<https://digitalcommons.usu.edu/cgi/viewcontent.cgi?article=3313&context=smallsat>

Staehle, Robert et al, “Interplanetary Cubesats: Opening the Solar System to a Broad Community at Lower Cost” Cubesat Workshop, 2011, Logan Utah.

Stakem, Patrick H., Kerber, Jonathas, “Rad-hard Software, Cubesat Flight Computer Self-monitoring, Testing, Diagnosis, and Remediation, 2017. available from the author.

Stakem, Patrick H. “Lunar and Planetary Cubesat Missions,” March Volume 15, Polytech Revista de Tecnologia e Ciência, avail: [http://www.polytech.com.br/revista\\_online/ed\\_15.pdf](http://www.polytech.com.br/revista_online/ed_15.pdf)

Stakem, Patrick H.; Da Costa, Rodrigo Santos Valente; Rezende, Aryadne; Ravazzi, Andre, Chandrasenan, Vishnu “A Cubesat-based Alternative for the Juno Mission to Jupiter,” 2017, Presented at the Flight Software Conference, FSW-17; avail:  
[http://flightsoftware.jhuapl.edu/files/\\_site/workshops/2017/](http://flightsoftware.jhuapl.edu/files/_site/workshops/2017/)

Stakem, Patrick H. *Interplanetary Cubesats*, PRRB Publishing, 2017, ISBN-1520766173.

Stakem, Patrick H. *Cubesat Constellations, Clusters, and Swarms*, PRRB Publishing, 2017, ISBN-1520767544.

Stakem, Patrick H. *Cubesat Engineering*, PRRB Publishing, 2017, ISBN-1520754019.

Stakem, Patrick H. *Cubesat Operations*, PRRB Publishing, 2017, ISBN-152076717X.

Stakem, Patrick H. “Free Software in Space—the NASA Case,” invited paper, Software Livre 2002, May 3, 2002, Porto Allegre, Brazil.

Stakem, Patrick H. “Lunar and Planetary Cubesat Missions,” March Volume 15, Polytech Revista de Tecnologia e Ciência,  
avail: [http://www.polytech.com.br/revista\\_online/ed\\_15.pdf](http://www.polytech.com.br/revista_online/ed_15.pdf)

Tan, Ying GPU-based *Parallel Implementation of Swarm Intelligence Algorithms*, 2016, 1st ed, Morgan Kaufmann, ISBN-978-0128093627.

Truskowski, Walt; Sterrotta, Roy; Rouff, Christopher A.; Hinchey, Michael G.; Rash, James L. “Next generation system and software architectures, Challenges from future NASA exploration missions,” 2005, in *Science of Computer Programming* 61 (2006) 48–57.

Truskowski, Walt, et al *Autonomous and Autonomic Systems: With Applications to NASA Intelligent Spacecraft Operations and Exploration Systems*, Springer, 1st Edition 2009, ISBN-1846282322.

Truskowski, Walt; Clark, P. E.; Curtis, S.; Rilee, M. Marr, G. “ANTS: Exploring the Solar System with an Autonomous Nanotechnology Swarm,” J. Lunar and Planetary Science XXXIII (2002).

Truskowski, Walt “Prototype Fault Isolation Expert System for Spacecraft Control,” N87-29136,  
avail: <https://ntrs.nasa.gov/search.jsp?R=19870019703>,

Violette, Daniel P. “Arduino/Raspberry Pi: Hobbyist Hardware and Radiation Total Dose Degradation, EEE Parts for Small Missions,” GSFC, 2014, avail: <https://ntrs.nasa.gov/search.jsp?R=20140017620>.

Virgili, Bastide et al “Mega-constellations Issues,” 41st COSPAR Scientific Assembly, 2016, avail: <http://cospar2016.tubitak.gov.tr/en/>

## Resources

- Ranging codes: <http://www.amsat.org/amsat/articles/g3ruh/123.html>
- Planetary Science Vision Workshop - <https://www.lpi.usra.edu/V2050/>
- Ball Aerospace Cosmos Product, [www.cosmosrb.com](http://www.cosmosrb.com)
- NASA, <https://nssdc.gsfc.nasa.gov/planetary/factsheet/asteroidfact.html>
- Small Spacecraft Technology State of the Art, NASA-Ames, NASA/TP2014-216648/REV1, July 2014.
- Core Flight System (CFS) Deployment Guide, Ver. 2.8, 9/30/2010, NASA/GSFC 582-2008-012.

- <http://sen.com/news/cubesats-set-for-new-role-as-planetary-explorers>

Author contact: [pstakem@gmail.com](mailto:pstakem@gmail.com)